

Create a graph database for students course registration create student and dept node and insert values of properties.

```
Create (n: student { sid: "VTU14500",  
  sname: "John",  
  deptname: "CSE" })
```

Output

Added 1 label, create 1 node, set 3 properties  
completed after 232ms

```
Create (n: student { sid: "VTU14501",  
  sname: "Dharsane",  
  deptname: "EEE" })
```

Output

Added 1 label create 1 node, set 3 properties,  
completed after 16ms

```
Create (n: student { sid: "VTU14502",  
  sname: "vijay",  
  deptname: "CSE" })
```

Output:

Added 1 label, created 1 node, set 3 properties,  
completed after 12ms.

## Task 11: CRUD Operation in Graph database

Aim:

To perform CRUD operation like creating, inserting, query, finding, deleting operation, on graph space.

Creating Node with properties.

Properties are the key-value pairs using which a node stores data. You can create a node with properties using CREATE clause. You need to specify these properties separated by commas within the flower braces "{}".

Syntax:

Following is the syntax to create a node with properties

```
CREATE (node: label {key1: value, key2: value, ...})
```

Returning the created Node

To verify the creation of the node, type and executed the following query in the Cypster prompt

```
MATCH (n) RETURN n
```

Creating Relationships

We can create a relationship using the CREATE clause. We will specify relationship within the square brace "[]" depending on the direction of the relationship it is placed b/w hyphen "-" and arrow "→" as show in the following syntax



select all nodes in your database using  
match command

match(n) return(n)

Output:

Vijay

Dharsana

CSE

John

match(n: student) return(n)

Output:

Vijay

Dharsana

John

a) Create relationship b/w student and CSE

MATCH (s: student), (d: dept) WHERE s.name = 'Vijay'

AND d.deptname = 'CSE'

CREATE (s) - [ST: STUDIED-AT] -> (d)

return s, d

Output:

CSE

Student AT

Vijay

Syntax:

Following is the syntax to create a relationship using create clause

```
CREATE (node1):[Relationship Type]->(node2)
```

Creating a Relationship Btw the Existing Nodes.

You can also create a relationship b/w the existing node using the MATCH clause

Syntax:

Following is the syntax to create a relationship using the MATCH clause

```
MATCH (a:Label Node 1).(b:Label Node 2)
```

```
WHERE a.name = "name of node 1" AND b.name = "name of node 2"
```

```
CREATE (a)-[:Relation]->(b)
```

```
RETURN a, b
```

Deleting a Particular Node

To delete a particular node, you need to specify the details of the node in the place of "n" in the above query.

Syntax:

Following is the syntax to delete a particular node from Neo4j using the DELETE clause.

```
MATCH (node: label {properties --- })
```

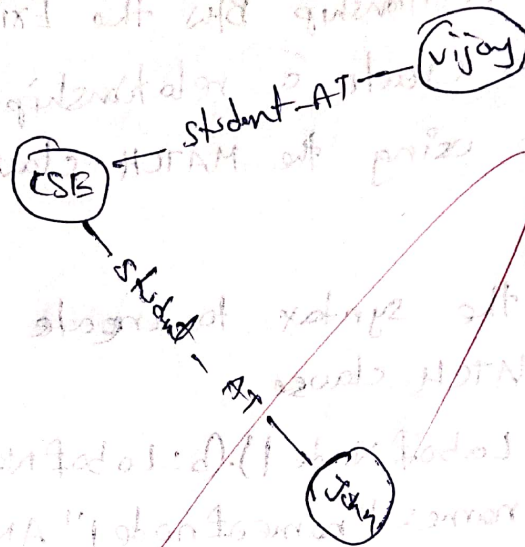
```
DETACH DELETE node.
```



b) Delete a node from student  
match (n: student { \$name: 'Dharmare' }) DELETE (n)

Output:

Deleted 1 node, completed after 15834 ms



VELTECH	
EX No.	11
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	3
VIVA VOCE (3)	3
RECORD (4)	4
TOTAL (15)	15
SIGN WITH DATE	

Result:

The implementation of CRUD operations like, inserting, finding and removing operations using GraphDB is successfully created.

13/06/24